(Preprint) AAS 10-188

FAILURE BOUNDING AND SENSITIVITY ANALYSIS APPLIED TO MONTE CARLO ENTRY, DESCENT, AND LANDING SIMULATIONS

John A. Gaebler[*] and Robert H. Tolson[†]

In the study of entry, descent, and landing, Monte Carlo sampling methods are often employed to study the uncertainty in the designed trajectory. The large number of uncertain inputs and outputs, coupled with complicated non-linear models, can make interpretation of the results difficult. Three methods that provide statistical insights are applied to an entry, descent, and landing simulation. The advantages and disadvantages of each method are discussed in terms of the insights gained versus the computational cost. The first method investigated was failure domain bounding which aims to reduce the computational cost of assessing the failure probability. Next a variance-based sensitivity analysis was studied for the ability to identify which input variable uncertainty has the greatest impact on the uncertainty of an output. Finally, probabilistic sensitivity analysis is used to calculate certain sensitivities at a reduced computational cost. These methods produce valuable information that identifies critical mission parameters and needs for new technology, but generally at a significant computational cost.

INTRODUCTION

In the study of entry, descent, and landing (EDL), Monte Carlo sampling (MCS) methods are employed to study the effects of input variable uncertainties on trajectory output variables. The MCS method involves the assembly of input cases by generating thousands of random input samples using estimated probability distributions. These input cases are then simulated with an EDL simulation yielding numerous output cases, one for each input case. Statistics are then accumulated for the output variables. Analyzing and gaining physical insight from these output statistics is often a difficult task given that there are often many input and output variables.

Complications can arise when there are failed cases that need to be understood. An example of such a failed case may be parachute deployment at an altitude below some minimum allowable value, which may lead to a reduced time between parachute deployment and landing. Another situation that can arise is a request for certain mission requirements, such as a smaller landing footprint, so that a lander will be close to a location of high scientific value. Reducing the uncertainty in an input can be a costly endeavor as it may require adding additional capability, developing new technology, etc. Given these issues, methods are sought that can provide insights into the statistical sensitivities of simulation outputs with respect to inputs of the EDL trajectories.

[*] Aerospace Engineer, NASA Goddard Space Flight Center, Code 595, Greenbelt, MD 20771.
[†] Professor, North Carolina State University, 100 Exploration Way, Hampton, VA 23666.

Three methods that provide statistical insights were identified and applied to an EDL simulation. The first technique studied was failure domain bounding.[1,2] If during an analysis a case fails it is imperative to understand the cause. This method can allow the generation of more failed cases with less simulations than typical MCS. Next a global variance-based sensitivity analysis was tested.[3,4,5] This method identifies which input variable uncertainty will have the greatest impact on reducing the variance on an output. This method has the additional benefit of identifying which inputs are interacting. Finally a method that provides local probabilistic sensitivities was studied.[6] These are sensitivities of an output mean or variance with respect to an input mean or variance. Leibniz's rule is introduced to evaluate certain partial derivatives with the benefit of requiring fewer simulations than finite differencing. These benefits are realized when calculating the sensitivities to uncorrelated inputs with probability distributions having infinite bounds. The advantages and disadvantages of each method are discussed in terms of the insights gained versus the computational cost.

EDL SIMULATION

A simplified simulation was needed that would be representative of an EDL mission, yet require less computer resources than a full simulation. The simulation created for the present study is a simplified model of a lander mission encompassing all phases of EDL including atmospheric entry, a parachute aided descent, and a thruster controlled landing. The equations of motion were limited to two dimensions and integrated with a Runge-Kutta fourth order algorithm.[*] The two dimensional ballistic EDL simulation included a temperature dependant atmosphere model, a rotating atmosphere with wind, and correlated entry states configured for Mars.

EDL Simulation Inputs

The EDL simulation had seventeen uncertain input variables that can be organized into three groups: initial states[7], vehicle properties[8,9], and atmospheric parameters[10,11]. The most complicated inputs from an analysis standpoint are the correlated initial states. Atmospheric variables have large uncertainties due to limited direct measurements of the Martian environment. Table 1 shows for each uncertain input the uncertainty and the probability distribution function (PDF) used. Uncertainty for a normal PDF is the 3σ value, while for a uniform PDF it is the bounds. The uncertainties of the initial states are omitted since they are correlated based on a prescribed covariance matrix.

EDL Simulation Outputs

Outputs from the EDL simulation include the states, vehicle mass, heat rate, total heat load, parachute deployment altitude, parachute deployment maximum acceleration, and propellant mass. After reviewing the output statistics from the simulation, six failure constraints were placed on five of the outputs. The constraints were arbitrarily chosen to give a low probability of failure, around 1.37% (i.e., a constraint violation of any one output constitutes failure).

[*] The derivation of the equations of motion, the required models, and the definition of the example system were provided by Dr. Juan R. Cruz.

Table 1. Uncertain inputs to the EDL simulation.

	Input	Symbol	Uncertainty	PDF
Initial States	inertial position	r_{IO}	N/A	Normal
	inertial speed	V_{IO}	N/A	Normal
	inertial flight path angle	γ_{IO}	N/A	Normal
	inertial longitude	λ_{IO}	N/A	Normal
Vehicle Parameters	drag coefficient of lander	C_{dae}	± 5 %	Normal
	drag coefficient of aeroshell	C_{dae}	± 5 %	Normal
	drag coefficient of parachute	C_{dp}	± 15 %	Uniform
	parachute opening load factor	C_X	± 10 %	Uniform
	terminal descent rocket engine specific impulse	I_{SP}	± 10 %	Normal
	entry mass	m_E	± 0.2 %	Normal
	parachute reference area	S_P	± 3 %	Uniform
Atmospheric Parameters	wind speed at surface	V_{wo}	± 100%	Uniform
	wind speed in upper atmosphere	V_{wiso}	± 60 %	Uniform
	altitude above which wind has constant speed	h_{Vwiso}	± 16.7 %	Uniform
	atmospheric density at surface	ρ_0	± 15 %	Normal
	atmospheric surface temperature	T_0	± 6.7 %	Normal
	atmospheric isothermal temperature above 57 km	T_{57}	± 12.9 %	Normal

FAILURE DOMAIN BOUNDING

To estimate the failure probability MCS can be applied. However, if the probability of failure is very small, MCS can be computationally expensive. Instead of sampling the entire input space, failure domain bounding (FDB) identifies an upper bound on a risky region.[1] Only cases within this risky region need to be sampled when approximating the probability of failure. Sample sets generated in the complement success region do not need to be simulated as they can be assumed to have only successful output cases (i.e. non-failures). Simulating input cases only from the risky region is an efficient means of generating failed cases for study and to increase the confidence in the failure probability estimation.

The following is a brief description of the FDB methodology. To find the upper bound on the risky region, the uncertain input variables PDFs are mapped into standard normal distributions with a mean of zero and variance of one. An optimization routine is then used to find the most probable point (MPP) of failure, which represents the point of the failure domain that is 'closest' to the nominal design. The distance from the nominal design to the MPP can be interpreted as the radius of a hypersphere.

Figure 1 illustrates this idea with a two dimensional example. The hatched area is the failure region. The design space within the hypersphere can be considered a success region, while everything beyond is a risky region that could have a failure. After the optimizer locates the MPP and the radius of the hypersphere, the failure probability can be estimated by sampling outside the hypersphere as represented by the dots. Typically FDB is applied to scenarios with a very small probability of failure.

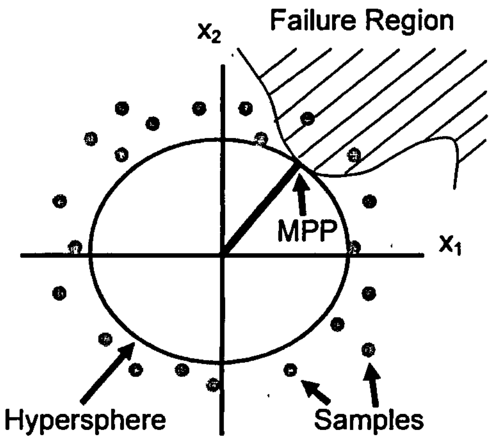

Figure 1. Example of success region defined by hypersphere radius.

A MCS analysis can now be performed on the parameters in the space beyond the hypersphere radius by not simulating cases generated within the hypersphere radius[1] or by conditional sampling, so as to never generate cases within the radius[12]. This method has the potential to greatly reduce the number of model evaluations needed to estimate the probability of failure, by assuming cases within the hypersphere are successful. Ignoring cases generated within the hypersphere frees up resources for more cases to be simulated within the risky region. Thus a better approximation of the failure probability can be obtained with fewer computations.

Application to EDL Simulation

Table 2 shows the radius for each constraint as they are optimized separately. The final result is the minimum hypersphere radius of all the constraints. Application of this method returned a hypersphere radius of 2.5541σ. Also shown is the number of simulations the optimizer required. The high number of simulations is caused by the large number of inputs and the use of a gradient based optimizer. Each time the optimizer takes a step toward the optimum, 18 simulations are required, once at the current point, and 17 times to perturb each input.

Table 2. Hypersphere radius found for each constraint in the EDL simulation

Constraint	Radius	Simulations
Peak heating > 26 W/cm^2	2.5729σ	1705
Altitude at parachute deployment > 9.5 km	2.7949σ	859
Maximum acceleration at parachute deployment > 13 g	2.5441σ	221
Final longitude < 14.5°	2.6970σ	348
Final longitude > 15.5°	3.2909σ	909
Percentage fuel consumed > 100%	2.6307σ	1712
Final	**2.5441σ**	**5754**

Each radius presented in Table 2 is the Euclidian norm of a vector pointing to the MPP of failure for each constraint. Plotting the components of this vector for the MPP associated with the altitude at parachute deployment as a bar graph gives Figure 2. This plot indicates which variables are affecting failure. The inputs defining the upper atmosphere and the drag force acting on the vehicle are the most influential. Those variables that have no effect on parachute deployment altitude are zero, such as the I_{SP}. From the equation for drag, the sensitivity to ρ_0 and $C_{d_{ae}}$ should be equivalent. The sensitivities differ in Figure 2 by a factor of three because ρ_0 has an uncertainty value of ±15% that is three times greater than the uncertainty on C_{dae} at ±5%.

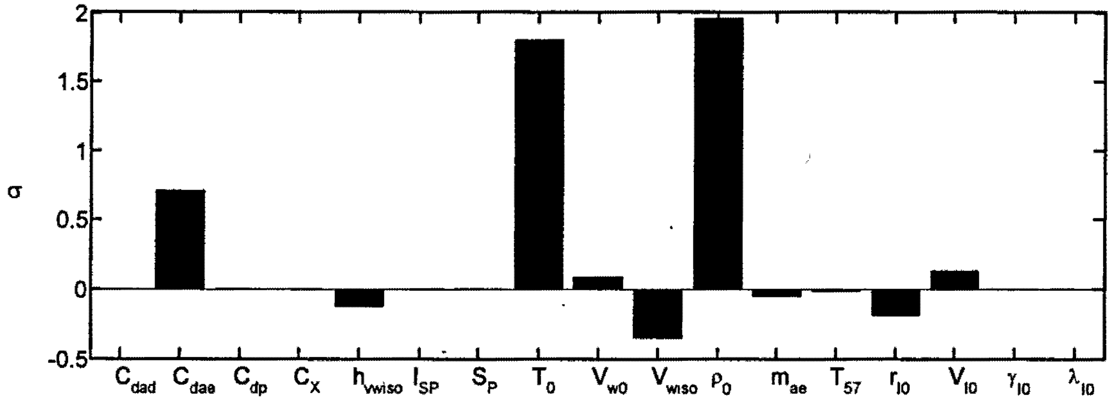

Figure 2. Vector components to the MPP of failure from parachute deployment altitude.

The next step is to start generating input cases and identifying those cases that can be assumed successful. Generating 30,000 random input cases gave 309 cases in the success region, which is only ~1%. The remaining 29,691 risky cases had 416 failures. There are no benefits to using this method since 99% of the design space must still be simulated. The percentage of failed cases generated increases by the inverse of the percentage of cases in the risky region.

Observations

An investigation into the short comings of FBD applied to the EDL simulation identified two complications. First was the assumption that a hypersphere, which encompasses the success region, would contain a significant portion of the design space. The second complication was with the minimization of the hypersphere over all inputs.

Further investigation of the first complication shows that the hypersphere does not contain a significant portion of the input design space due to how a large number of input dimensions spread the design space PDF. This spreading undermines the usefulness of minimizing a hypersphere around the nominal design. Since the inputs have standard normal distributions, a radius around the origin should encompass a large portion of the sample set. For example a standard normal distribution of a single variable with a hypersphere radius of 2σ would encompass ~95% of the design space. With two variables, the 2σ circle only contains ~86% of the design space. A third variable reduces the 2σ sphere to 74%. This trend continues as more dimensions are considered.

The second complication involves minimization of the hypersphere radius vector over all inputs. The vector component for any input with minimal influence on the failure region will have a distance near zero. However this input can have any value without affecting failure. Thus including the component unnecessarily reduces the number of cases in the success region.

To get around this complication, components of the output hypersphere radius vector with values less than some tolerance were ignored in defining the success region. It is desirable to ignore insensitive inputs since it increases the success region, while also reducing the dimensionality of the problem alleviating the spreading effect. When identifying whether a case is a success or not, only the inputs affecting failure are considered. However, now each constraint must be considered in defining the success region. An input that does not influence one failure constraint may be important to another. If that variable were ignored there could be failed cases in the success region due to the other constraints. In essence this new approach identifies separate success regions, or individual hyperspheres, for each constraint, where each hypersphere has minimal dimensions.

The original methodology would have had a single 17 dimensional hypersphere radius that should in theory touch the MPP of failure. The new methodology proposed has six hyperspheres, of only 4-6 dimensions each. The tolerance used to ignore a component was set to 0.2σ. Only cases common to all six hyperspheres are considered successful. The effect is to increase the success region from containing 1% to over 42% of the sample set. Of the 30,000 simulations, no failed cases were identified in the success region at this tolerance. Figure 3 shows the convergence of the estimated failure probability by performing typical MCS versus using FDB. The FDB plot starts at 5754 model runs, since those were required by the optimizer to find the MPP. Both methods have similar convergence. The benefits would increase if the dimensionality of the hyperspheres could be reduced further, or if there were less failure constraints.

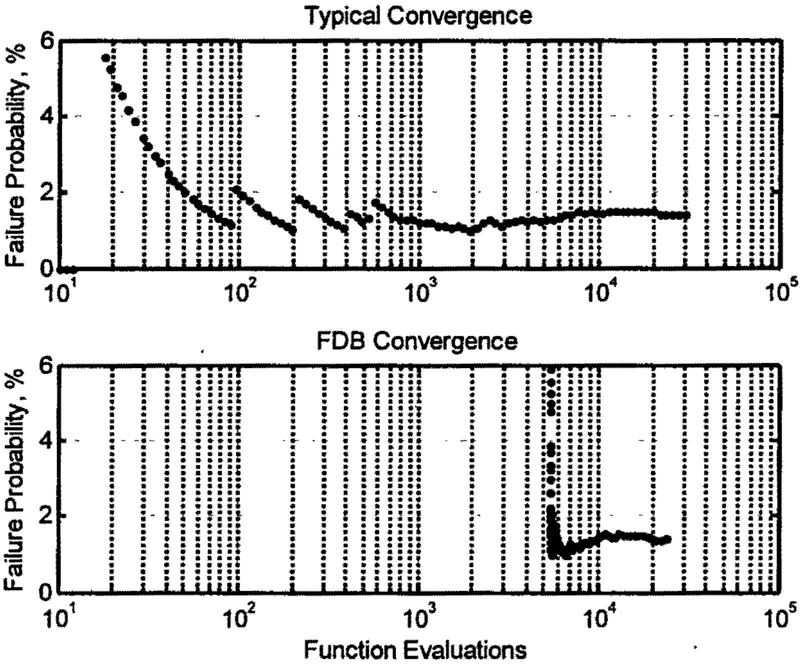

Figure 3. Convergence of EDL simulation failure probability for MCS vs. FDB.

GLOBAL VARIANCE-BASED SENSITIVITIES

In the literature this method is often referred to as Sobol's method. It provides a way of representing a global type of sensitivity based on the variance of the output.[3] Advances to the original method that reduce the required computations are incorporated in this work.[4] Global sensitivity measures the effect input variance, for a specific input V_i or combination of inputs V_{ij}, has on the output variance. This method works on the premise that the total variance of an output variable can be segmented into components caused by individual or combinations of input variables. Thus allowing the comparison of the variance due to any input variable with the total variance on the output, giving a sensitivity as shown in Equation (1).

$$S_i = \frac{V_i}{V} \text{ or } S_{ij} = \frac{V_{ij}}{V} \tag{1}$$

The sensitivity can be found for each input separately or in any combination with other input variables. Sensitivities due to individual variables are considered main or 1st order effects. Coupling or 2nd order effects comprise all possible combinations of two variables. Combinations of variables can be up to order n, where n is the total number of input variables. All the sensitivities sum to a value of one. Stated another way, the portion of the output variance attributed to each input and combination of inputs must add up to the total variance. Another parameter of interest is the total effect sensitivity.[13] Summing sensitivities, such as the main effect and all higher order terms involving the variable of interest, gives the total effect. These ideas are portrayed in Equations (2) and (3) for a 3 input model, where the number of subscripts represents the order (1st, 2nd...) and the superscript T is for total effect.

$$S = 1 = S_1 + S_2 + S_3 + S_{12} + S_{23} + S_{13} + S_{123} \tag{2}$$

$$S_1^T = S_1 + S_{12} + S_{13} + S_{123} \tag{3}$$

Conceptually Figure 4 shows what type of information Sobol's method provides. The left plot shows the output of a function with two inputs sampled 20 times. When the first input set is resampled, while leaving the second set the. same, the center plot is generated. Hence each output would change since one of the inputs is different. Next the right plot is produced by resampling the second variable and using the original sample for the first. Again the outputs would change. The original outputs were left in the plot for reference. By taking this new variance and comparing it to the original, an idea of the influence from each variable has is obtained. In Figure 4 it is obvious that the second variable has a larger effect on the output variance. Hence the second variable would have a larger sensitivity.

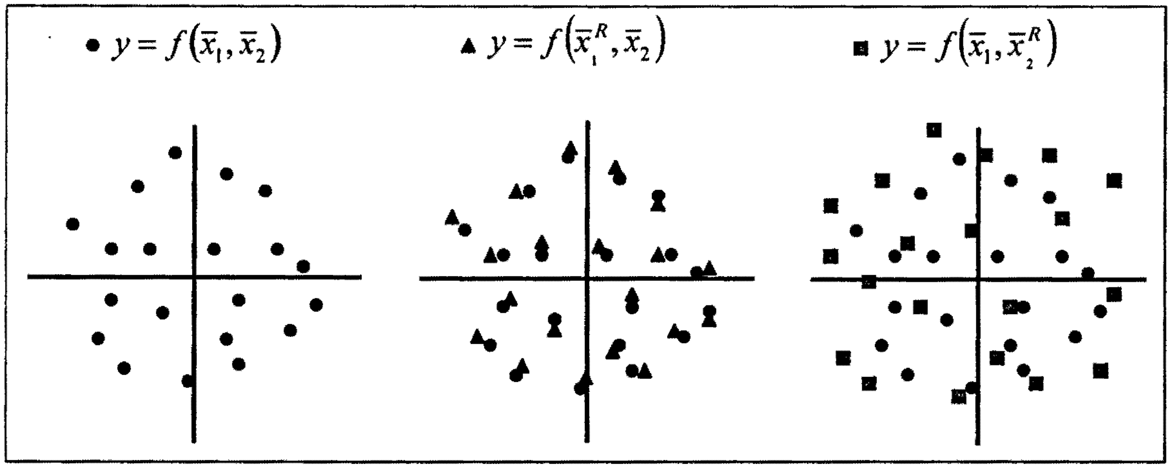

Figure 4. Visualization of Sobol's method, resampled sets have superscript R.

To isolate the variance of a specific input i all the inputs are varied via MCS to approximate the total variance V, then the complimentary variables $j \neq i$ are resampled to approximate the variance caused by i. The required integrals for calculating the variance are approximated by MCS. The equations in approximate form are:

$$f_0 \approx \frac{1}{N}\sum_{i=1}^{N} f(\bar{x}) \tag{4}$$

$$V + f_0^2 \approx \frac{1}{N}\sum_{i=1}^{N} f^2(\bar{x}) \tag{5}$$

$$V_i + f_0^2 \approx \frac{1}{N}\sum_{i=1}^{N} f(\bar{x}) f(\bar{x}_i, \bar{x}_j^R) \tag{6}$$

Here $\bar{x}$ is an array of the n inputs with N samples and $\bar{x}_i$ is the 1 by N set of samples of the input for which the sensitivity is desired. Finally $\bar{x}_j^R$ is the set of (n-1) by N samples that are resampled. The function f( ) represents the simulation with f_0 being the mean of the output variables. Choice of sampling method can be important. Low discrepancy sequences, or quasi-random number generators, get better results than pseudo-random number generators by picking numbers in a manner that attempts to guarantee uniformity across the design space.[14] Two such quasi-random number generators available to use are Halton sequence leaped[14] and Sobol' sequence generators[*].

The most influential variable can be identified by comparing the individual effects of each variable on the output variance. Such knowledge helps in deciding where to focus efforts to reduce the uncertainty on the output to obtain the most benefit. Considering combinations of variables, as opposed to individual variables, gives an indication of the coupling between inputs.

One drawback is that a large sample set is required to obtain reliable statistics. Each input requires N runs times the number of terms involving that input sought. Equations (7) and (8) show the simulations required to obtain the desired sensitivities where T is the number of terms sought. Thus for each individual or coupled sensitivity desired, the model must be evaluated with a separate sample set.

$$\text{main effects cost} = N(T+1) \tag{7}$$

$$\text{main} + \text{total effects cost} = N(T+2) \tag{8}$$

There are some requirements on the input parameters such as the inputs must be uncorrelated and transformed to uniform distributions on the unit hypercube. In addition, to evaluate the accuracy of the results, there are methods to obtain confidence intervals.[5]

[*] Available at http://www.broda.co.uk accessed September 2008.

Application to EDL Simulation

Due to the large number of inputs for the simulation, which increases the samples needed for convergence, only main effects and total effects were sought. To give an idea of the sensitivities obtained, one output will be investigated. Figure 5 shows the main effect or 1st order sensitivities of the parachute deployment altitude.

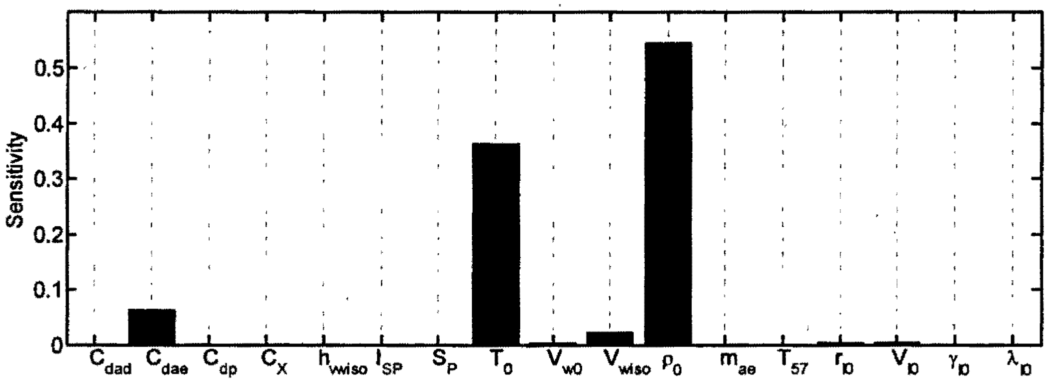

Figure 5. Global sensitivities on parachute deployment altitude.

The uncertainty in the parachute deployment altitude comes mainly from the surface density and temperature which determine the atmospheric model. These terms affect the density in the upper atmosphere, which affects the drag acting on the vehicle. The entry configuration drag coefficient is the other primary contributor. The values make physical sense and match the two analyses presented thus far. In this case however, the sensitivity to ρ_0 is roughly nine times higher than that on C_{dae}, due to this method giving sensitivities based on the variance ($V = \sigma^2$).

A large number of runs were required to obtain these sensitivities. Figure 6 shows an example of the convergence for the sensitivity of parachute deployment altitude to surface temperature. If based on this plot N is chosen to be 6,000, then using Equation (7) with $T = 17$ inputs, it would take 108,000 runs to obtain the main effect sensitivities. An additional 6,000 runs will provide the total effect sensitivity terms, which should indicate which variables have higher order couplings. An investigation of these terms does not indicate any significant couplings for this output.

Observations

Sobol's method had difficulty approximating the sensitivities to outputs with small output variances. Less than half of the output variables converged on a solution. If convergence to an accurate sensitivity isn't required, but instead the relative sensitivities are acceptable, a smaller sample set could be run. After $N = 2{,}000$ runs (for each sensitivity) the top three contributors remained relatively constant. For example the top three contributors to the maximum acceleration at parachute deployment were C_{dp}, C_X, and S_P in that order. Increasing N does not change that order. The actual sensitivity values haven't converged in 2,000 runs, but the relative values have.

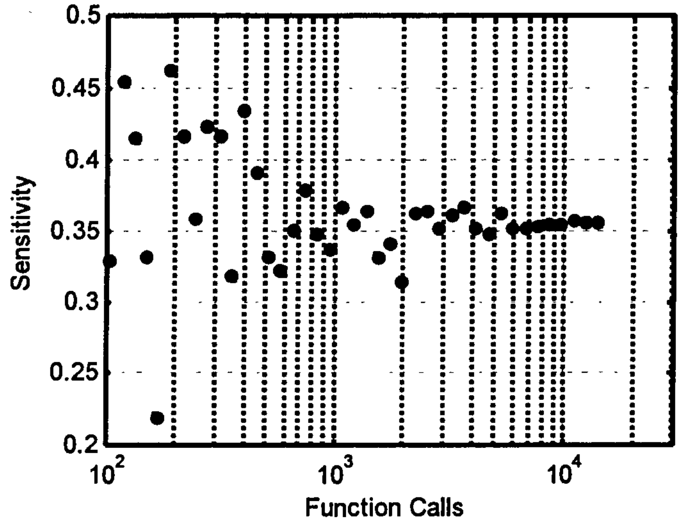

Figure 6. Parachute deployment altitude sensitivity convergence to surface temperature.

LOCAL PROBABILISTIC SENSITIVITIES

The goal of the probabilistic sensitivity analysis (PSA) is to approximate the partial derivatives of the output statistics with respect to the input statistics.[6] The results of PSA can be considered local in that they are evaluated at a particular value, as compared to Sobol's method which considered the whole range of the statistics. These probabilistic sensitivities are found by taking the partial derivative of the mean or variance of the output with respect to the mean or variance of a specific input variable: $\partial\mu_{out}/\partial\mu_{in}$, $\partial\mu_{out}/\partial v_{in}$, $\partial v_{out}/\partial\mu_{in}$, or $\partial v_{out}/\partial v_{in}$.

Calculation of the mean or variance can be found with the expected value operator in Equation (9), where $y(\ )$ is an output function and $\overline{X}$ is the array of input parameters with known PDFs represented jointly as $p(\ )$. Two methods are proposed by Crespo[6] to obtain the sensitivities, the first of which is finite differencing of the expected output in Equation (10), where θ_i is either the mean, μ_{x_i}, or variance, v_{x_i}, of the input vector $\overline{x}_i$ for a single input *i*. Since $\overline{X}$ represents a sample set there is a mean and variance associated with each input. The variable of interest $\overline{x}_i$ can be transformed to reflect a perturbation of Δ to the μ_{x_i} or v_{x_i}, which is incorporated into $\overline{X}_\Delta$ with all other inputs unchanged.

$$E\left[y(\overline{X})\right] = \int y(\overline{X})p(\overline{X})d\overline{X} \tag{9}$$

$$\frac{\partial E\left[y(\overline{X})\right]}{\partial\theta_i} \approx \frac{1}{\Delta}\left(E\left[y(\overline{X}_\Delta)\right] - E\left[y(\overline{X})\right]\right) \tag{10}$$

The second method applies Leibniz's rule to find the derivative of the integral in the expected value operator in Equation (9). Calculating the sensitivity requires moving the derivative inside the integral and simplifying to Equation (11). In this equation $p_i(\ )$ is the PDF for $\bar{x}_i$. The multidimensional integral in the expected value operator in Equation (9) is integrated over the bounds of each input PDF. Equation (11) includes the bounds of $\bar{x}_i$ as a and b. Sample set $\bar{X}_a$ has the input vector $\bar{x}_i$ replaced with the boundary value a for all samples, while leaving all other variables unchanged. Similarly $\bar{X}_b$ has b replacing the input vector $\bar{x}_i$. It is assumed that the partial derivatives of $p_i(\ )$ can be found analytically.

$$\frac{\partial E[y(\bar{X})]}{\partial \theta_i} = E\left[\frac{y(\bar{X})}{p_i(\bar{x}_i)}\frac{\partial p_i(\bar{x}_i)}{\partial \theta_i}\right] + \frac{\partial b}{\partial \theta_i} p_i(b) E[y(\bar{X}_b)] - \frac{\partial a}{\partial \theta_i} p_i(a) E[y(\bar{X}_a)] \tag{11}$$

At first glance there is no apparent benefit over finite differencing in terms of computational expense: $y(\bar{X})$ must be evaluated, then for each partial $y(\bar{X}_b)$ and $y(\bar{X}_a)$ must also be evaluated via numerical integration. However for PDFs with bounds at infinity, the second and third terms in equation (11) become zero (e.g. for a normal distribution $p(\infty) = 0$) reducing the number of cases simulated. If a set of cases are simulated to obtain the mean and variance of the outputs, e.g. $E[y(\bar{X})]$, the partial derivatives with respect to inputs having normal distributions can be obtained at no additional computational cost (i.e., no additional samples), since $y(\bar{X}_b)$ and $y(\bar{X}_a)$ need not be simulated. Similarly, equations can be defined for the sensitivity of the variance $V[y(\bar{X})]$:

$$\frac{\partial V[y(\bar{X})]}{\partial \theta_i} = E\left[\frac{(y(\bar{X}) - E[y(\bar{X})])^2}{p_i(\bar{x}_i)}\frac{\partial p_i(\bar{x}_i)}{\partial \theta_i}\right] + \frac{\partial b}{\partial \theta_i} p_i(b) V[y(\bar{X}_b)] - \frac{\partial a}{\partial \theta_i} p_i(a) V[y(\bar{X}_a)] \tag{12}$$

To provide a fair comparison between the derivatives, the outputs are given as percent derivatives. Calculation of percent derivatives requires defining step sizes. The step Δ is used as a percentage of the statistic to be perturbed. In this work Δ is set to 1% of the input variable mean if the partial derivative is with respect to the mean (similarly for the variance). If the input mean is zero, then Δ is taken as 0.5% of the standard deviation. When using Leibniz's rule the step size is used to scale the results of the method. Sample sets, for the calculation of the sensitivities, can be generated by several methods such as: Sobol sequence generator*, pseudo-random number generator, or Halton sequence generator[14].

Obtaining the sensitivity of an output to each input mean allows for the identification of the largest contributing input variable. Calculating the sensitivity to the variance on the inputs would show where the most benefits could be gained by performing further experimentation (such as analysis or hardware changes, etc) to reduce uncertainty on an input. Application of Leibniz's rule is promising because it requires the evaluation of only one sample set of N samples to obtain numerous sensitivities when the bounds of the input PDFs are infinite such as in a normal distribution.

* Available at http://www.broda.co.uk accessed September 2008

There are some limitations to this method. The analytic partial derivative of the PDF is required for the Leibniz approach. For this work there were no issues with deriving the analytic partial derivatives, however there may be PDFs where this can be a concern. While no additional simulations are required to obtain sensitivities for uncorrelated normal distributions beyond a first set, there is an additional computational cost for PDFs having bounds that are functions of θ, such as uniform distributions. Equation (13) shows the computations required where U is the number of input variables with PDFs having bounds dependent on θ. For the worst case scenario, where none of the input PDFs are normal, the computational cost of using Leibniz's rule is equivalent to that required for finite differencing when calculating also the possible terms. Therefore the entire analysis can be performed via the Leibniz's rule with the benefit of fewer simulations if there are input variables with normal distributions.

$$\text{Leibniz Cost} = N(2U+1) \tag{13}$$

Application to EDL Simulation

A large amount of information is provided by PSA. For each output there are four partial derivatives per input. Thus for the EDL simulation there are 68 partial derivatives for each output (4*17 inputs). The parachute deployment altitude output is analyzed here. Four bar charts are presented, one for each set of sensitivities.

First the sensitivities to the parachute deployment altitude mean with respect to the input means are shown in Figure 7. The initial states were omitted from the following plots because they were correlated, which will be discussed later. The sensitivities are scaled to show how a 1% change in the mean surface temperature causes a 2.63% change in the mean parachute deployment altitude for example. This plot suggests that the mean surface temperature, surface density, and initial mass have the greatest influence on the mean parachute deployment altitude. These results make physical sense since the first two terms dictate the atmospheric density, while the last term will control how fast the vehicle is traveling.

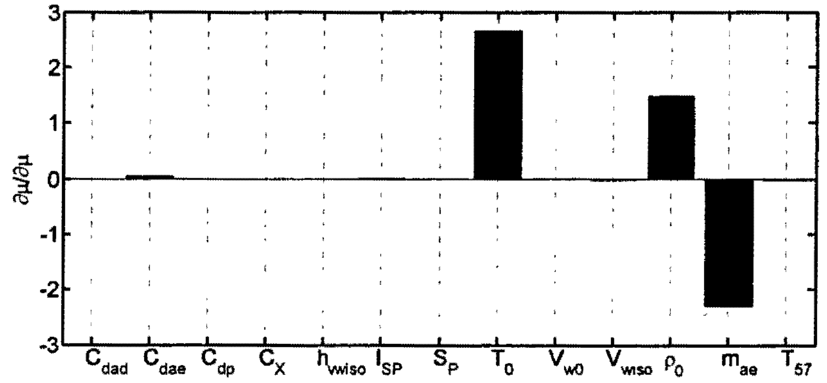

Figure 7. Sensitivity of parachute deployment altitude mean to each input mean.

Next the sensitivity of the output mean with respect to the input variance can be investigated. Figure 8 shows that a change to the variance of any input has little effect on the output mean. The largest term only causes a 0.002% change to the output mean. While the larger terms in this plot are those that define the upper atmospheric conditions, this plot appears to be influenced by numerical noise due to running a limited sample set.

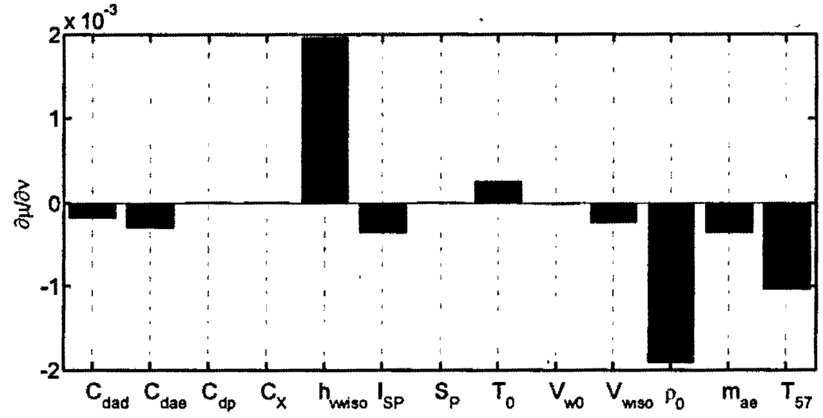

Figure 8. Sensitivity of parachute deployment altitude mean to each input variance.

The sensitivity of the output variance due to the input mean is shown in Figure 9. Here the major influence is from the initial mass with a small effect from the surface temperature. Increasing the mass of the vehicle should reduce the effect of the aerodynamic forces acting upon it.

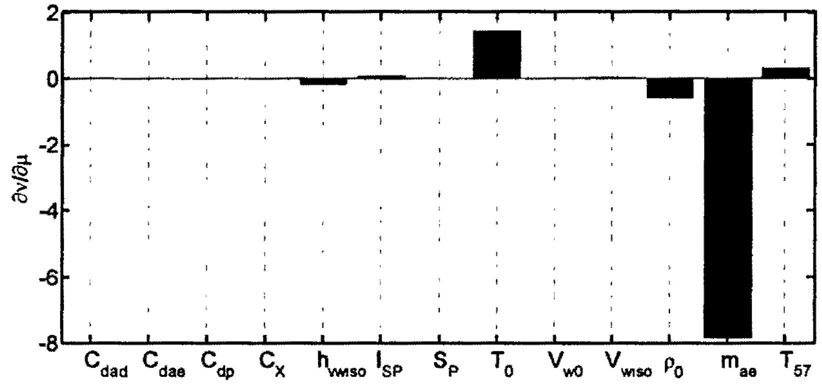

Figure 9. Sensitivity of parachute deployment altitude variance to each input mean.

Finally the output variance sensitivity to input variance is shown in Figure 10. Again the atmospheric variables, surface temperature and density, are the main contributors. In addition there is some influence from the coefficient of drag for the aeroshell. These results again match with those found by the previous two methods in this report. The sensitivity in this chart is based on the uncertainty, thus if there are different uncertainties assigned to the inputs C_{dae} and ρ_0, this variation will appear in the sensitivities. Again the sensitivity to ρ_0 is roughly nine times greater than that of C_{dae}.

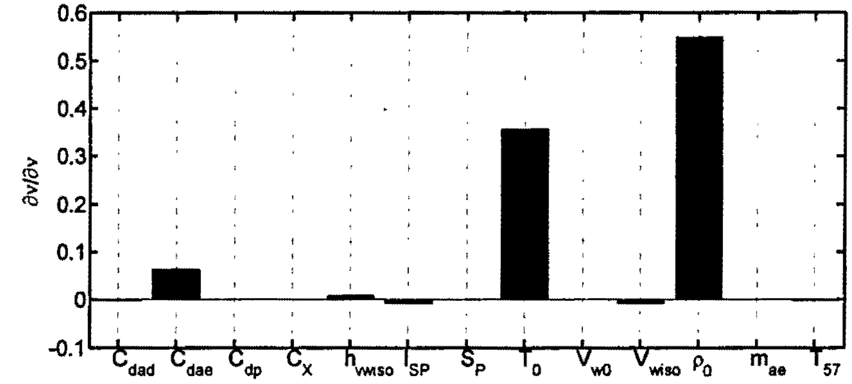

Figure 10. Sensitivity of parachute deployment altitude variance to each input variance.

To give accurate results, the sample size for these results was set at 15,000. Figure 11 shows an example of the typical convergence. Choosing N = 5,000 using Equation (13) with U = 6, since there are six inputs with uniform distributions, gives 65,000 total simulations. Calculating these sensitivities using finite differencing would have required 175,000 simulations to obtain the same results.

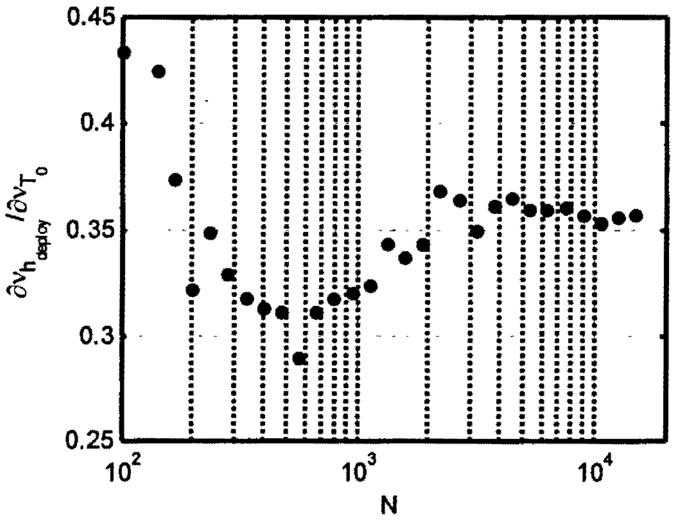

Figure 11. Convergence of parachute deployment altitude to surface temperature.

Observations

As previously mentioned, the initial states were ignored in this section. The analytic partial derivative of the PDF required by Leibniz's rule is difficult to calculate for correlated variables. Since the four ignored inputs are correlated, the joint PDF is difficult to separate when calculating the analytic partial derivatives as can be done with independent inputs. The results found match with what is calculated by finite differencing. Given a situation where partial derivatives of the statistics to an input set having several normally distributed inputs are desired, application of Leibniz's rule has clear advantages.

CONCLUSION

This study analyzed the merits of three methods that provide insights into the statistics of EDL simulations. The failure domain bounding method promised to provide a more accurate approximation of the failure probability with less runs than a typical MCS analysis. Global variance-based sensitivities and probabilistic sensitivity analysis are methods that provide the sensitivity of the output statistics to the input statistics. These three methods were applied to a 2D EDL simulation for assessment.

When the FDB method was applied to the EDL simulation an optimizer required 5,754 runs to find a hypersphere radius defining the success region. For the variables that dictate failure, the radius vector also gives a sense of the sensitivities to the inputs. The success region was found to encompass an insignificant portion of the design space due to a spreading effect of increasing the input dimensions. A modification was implemented to partially fix the problem. Instead of identifying one small success region, several regions are found, and then combined to give a larger success region. With the given failure constraints convergence to the probability of failure was equivalent to typical MCS. FDB gives better results when there are fewer constraints, smaller failure probability, or fewer input variables.

Sobol's method successfully gave the sensitivities to each input variable. Accurate results required 108,000 runs for main effect sensitivities. Variance-based sensitivities also have the advantage of providing sensitivities to the interactions between inputs. Inspection of the total effect sensitivities did not indicate many significant couplings for the output studied. It was found that the method did not work well if the output variance was small. This method works better with large output variances and less input variables.

The PSA method also successfully gave the sensitivities to the input statistics. While it does not provide a direct sense of couplings that may be present, it does provide sensitivities to both the mean and variance. It required 65,000 runs for accurate results, providing similar information to Sobol's method at a fraction of the simulations. PSA provides the same information as finite differencing, again with fewer simulations. The method requires the analytic partial derivatives of the PDFs with respect to the mean and variance, which was not an issue for the situation studied, but could be for other PDFs. Application of Leibniz's rule to variables that are correlated requires further study. Sensitivities of the failure probability or to correlated inputs can be obtained via finite differencing. The benefits of PSA are augmented with an increase in the number of input variables defined with normal distributions, or other distributions with infinite bounds.

ACKNOWLEDGEMENTS

This work was done in partial fulfillment of a Master of Science degree through North Carolina State University and supported under NASA cooperative agreement NCC1-02043.

REFERENCES

[1] L.G. Crespo, D.P. Giesy, S.P. Kenny, "Reliability-based Analysis and Design via Failure Domain Bounding" *Structural Safety*, Vol. 31, July 2009, pp. 306-315.

[2] L.G. Crespo, D.P. Giesy, S.P. Kenny, "Robustness Analysis and Robust Design of Uncertain Systems" *AIAA Journal*, Vol. 46, No. 2, 2008, pp. 388-396.

[3] I.M. Sobol', "Sensitivity estimates for nonlinear mathematical models" *Matematicheskoe Modelirovanie*, Vol. 2, No. 1, 1990, pp. 112-118 (in Russian), English translation in: *Mathematical Modeling and Computational Experiment*, Vol. 1, No. 4, 1993, 407-414.

[4] A. Saltelli, "Making Best Use of Model Evaluations to Compute Sensitivity Indices" *Computer Physics Communications*, Vol. 145, 2002, pp. 280-297.

[5] G. Archer, A. Saltelli, I.M. Sobol', "Sensitivity measures, ANOVA like techniques and the use of bootstrap" *Journal of Statistical Computations and Simulation*, Vol. 58, 1997, pp. 99-120.

[6] L.G. Crespo, S.P. Kenny, D.P. Giesy. "Sampling-based Strategies for the Estimation of Probabilistic Sensitivities", *AIAA Conference in Non-deterministic Approaches*, Palm Springs, California, 4-7 May 2009. AIAA-2009-2283.

[7] R.D. Braun, R.M. Manning, "Mars Exploration Entry, Descent, and Landing Challenges", *2006 IEEE Aerospace Conference*, IEEEAC paper #0076, Big Sky, Montana, March 2006.

[8] W.W. Blake, "Experimental Aerodynamic Characteristics of the Viking Entry Vehicle Over the Mach Range 1.5-10.0", NASA CR-159225, 1971.

[9] P. Desai, M. Schoenenberger, F.M. Cheatwood, "Mars Exploration Rover Six-Degree-of-Freedom Entry Trajectory Analysis", *Journal of Spacecraft and Rockets*, Vol. 43, No. 5, Sept-Oct 2006.

[10] D. Kaplan, *Environment of Mars*, 1988, NASA TM-100470, 1988.

[11] A. Seiff, "Post-Viking Models for the Structure of the Summer Atmosphere of Mars", *Adv. Space Res.* Vol. 2, 1982, pp. 3-17.

[12] D.P. Giesy, L.G. Crespo, S.P. Kenny, "Approximation of Failure Probability Using Conditional Sampling", *12th AIAA/ISSMO Multidisciplinary Analysis and Optimization Conference*, Victoria, Canada, 10-12 September 2008. AIAA 2008-5946

[13] I.M. Sobol', "Global Sensitivity Indices for Nonlinear Mathematical Models and their Monte Carlo Estimates", *Mathematics and Computers in Simulation*, Vol. 55, No. 1-3, 2001, pp. 271-280.

[14] L. Kocis, and W.J. Whiten, "Computational Investigations of Low-Discrepancy Sequences", *ACM Transactions on Mathematical Software*, Vol. 23, No. 2, June 1997, pp. 266-294.